International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Model of a Job Traffic Queue of a Cloud- Based Research Collaboration Platform

Hyacinth C. Inyiama[Ȧ] and Nkolika O. Nwazor[Ḃ*]

[Ȧ]Electronic and Computer Engineering Department, Nnamdi Azikiwe University, Awka, Anambra State Nigeria
[B]Electronic and Computer Engineering Department, University of Port Harcourt, Choba, Rivers State Nigeria

Accepted 05 Sept 2014, Available online 01 Oct 2014, **Vol.4, No.5 (Oct 2014)**

Abstract

This paper discusses the model of a Job Traffic Queue of a cloud based research collaboration platform. The objective is to compute the size of those queues, the time that jobs spend in them i.e. number of simultaneous HTTP GET file requests handled by the server, and the total time required to service a request. This system captures efficiency in service time, server utilization, queuing response times, queuing workload and contents for a Cloud based Research Collaboration platform. MATLAB Simevent tool 2009b was used to develop the cloud queuing model. The proposed Queuing-Model-Based Adaptive Control approach combines both the modelling power of queuing theory and self-tuning power of adaptive control. Therefore, it can handle both modelling inaccuracies and load disturbances in a better way.

Keywords: *Queuing, Architecture, collaboration, cloud computing, workload, Model, service time, Traffic*

1. Introduction

Queuing theory is the mathematical study of waiting lines, or queues. In queuing theory a model is constructed so that queue lengths and waiting times can be predicted (Wikipedia, 2013). The work in (Leung, 2000) sees queuing theory as the study of queuing systems, where some customers get some services from some servers. Queuing theory is useful in telecommunication for determining throughput, response time, utilization, lost call probability, and resource requirements to mention but a few. Through the prediction of the system, the regulation about the queuing probability can be revealed and the optimal method for the system can be chosen. Queuing theory can be adopted to estimate the network traffic, which becomes the important ways of network performance prediction, analysis and estimation (Jian-Ping and Yong, 2008). Through this way, the true network can be imitated. This is useful and reliable for organizing, monitoring and defending the network. The cloud-based research collaboration platform referred to as CRCM (Nwazor, 2014) is a platform that fosters innovation by incorporating knowledge management into research processes, allowing researchers to share information and solve research problems more efficiently. It incorporates virtual research equipments It is a platform that enables researchers to collaborate with each other and with the industrialists who are the end users of research outputs.

Scheduling policies that can be used at queuing nodes include (Wikipedia, 2013):

- *First in First out*: here customers are served one at a time and the customer that has been waiting the longest is served first
- *Last in First out:* here customers are also served one at a time, however the customer with the shortest waiting time will be served first.
- *Processor Sharing*: here service capacity is shared equally between customers
- *Priority*: Customers with high priority are served first. Priority queues can be of two types, non-pre-emptive (where a job in service cannot be interrupted) and pre-emptive (where a job in service can be interrupted by a higher priority job). No work is lost in either model
- *Shortest job first*: The next job to be served is the one with the smallest size
- *Pre-emptive shortest job first:* The next job to be served is the one with the original smallest size
- *Shortest Remaining Processing time:* The next job to be served is the one with the smallest remaining processing requirement

2. Framework for CRC Platform Queuing Model

This model is a generally applicable model that abstracts all hardware and software details, but is specific enough to produce significant performance results regarding the relationship between authenticated user processes from the collaborators, internet cloud, backend traffic and cloud server. In this work, this model took cognizance of the high-level details of the HTTP and TCP/IP protocols.

2.1 CRC Queuing Design Formulations

In the proposed design, a web server W_s together with a browser program (i.e., client) constitutes a client/server system as shown in fig. 1. W_s typically process many simultaneous jobs (i.e., file requests), each of which

*Corresponding author: **Nkolika O. Nwazor** contends for various shared resources: processor time, file access, and network bandwidth. Since only one job may use a resource at any time, all other jobs must wait in a queue for their turn at the resource.

As jobs receive service at the resource, they are removed from the queue; and as jobs arrive, they join the queue immediately. The CRCM Queuing model will compute the size of those queues and the time that jobs spend in them. The concern here is the number of simultaneous HTTP GET file requests handled by a server, and the total time required to service a request. Now, the CRCM views every service or resource as an abstract system consisting of a single queue feeding one server for various users.

Let the CRCM queue made by the users be Q. Associated with every queue are viz:

1) Arrival rate (A) —The average rate at which new jobs arrive at the queue.
2) Service time *(TS)* of the server - The average amount of time that it takes a server to process such jobs. $TS = \frac{1}{\mu}$
3) Queuing time *(*TQ) - The average amount of time a job spends in the queue. The average response time (T) is given by $TS = \sum TS + TQ$

   If the Arrival rate *A* is less than the service rate (1/TS), then the CRC queuing system is *stable* as such all jobs will eventually be serviced, and the average queue size is bounded. On the other hand, if *A > (1/TS),* then the system is *unstable* and the queue will grow without bound.
4) Server utilization *U* - This is the product of the arrival rate (*A)* and service time (TS). Hence, $f(U) = A * TS$

This is a dimensionless number between 0 and 1 for all stable systems. A utilization of 0 denotes an idle server, while a utilization of 1 denotes a server being used at maximum capacity. At a utilization of 0 (zero), the response time is just the service time; no job has to wait in a queue. As utilization increases, the response time of the queue grows gradually. Only when the utilization approaches 1 (one) does the response time climb sharply toward infinity. The web server queuing model satisfies this behaviour.

Assuming the amount of time between job arrivals *(1/A)* is random and unpredictable, then the arrivals exhibit an exponential or memoryless distribution.

Also, the service rate must be greater than the arrival rate, that is, $1/TS = \mu$ > *A*. If $\mu \leq A$ *then* the waiting line (FIFO queue) would eventually grow infinitely large.

2.2 Cloud Queuing Model Design

Fig. 1 shows the architecture of the cloud queuing model for the CRCM comprising of the internet and the web server network models which will be realized and analysed with a MATLAB Simevent tool showing the cloud queue metrics for CRCM. It serves as a simple file server over the collaboration internet. As shown in the figure, the proposed model was assumed to have two nodes in the internet communication network (*In_1, In_2*) and two nodes in the web server entity (*Ws_1, Ws2).* The cloud queue model also has the variant process controls as well as its background processes.

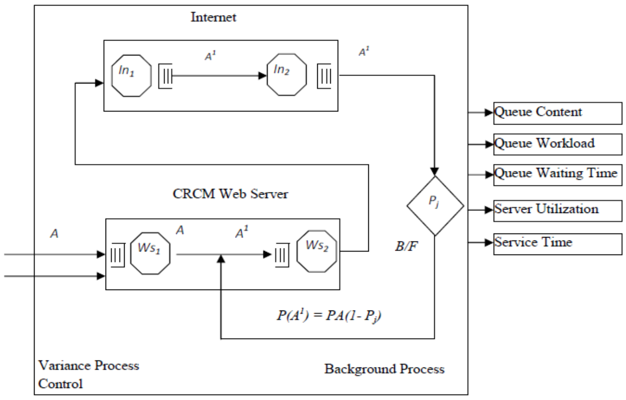

Fig. 1 Architecture model of Cloud Queuing for CRCM

Now, let all the file requests or jobs by users arrive at the web server with an arrival rate of *A.*

Let all the instance initialization processing be performed by node *$Ws_{1.}$* The job then proceeds to node *Ws_2* where a single buffer with data is read from the file, processed and passed into the internetwork at *$In_{1.}$* This block of data (FIFO) is then transmitted to the internet at the server's transfer rate. This data travels via internet and is received by the client browser respectively by node *$In_{2.}$* If the file was not completely transmitted, the job returns back to node *Ws_2* for further processing, else, the queuing job is complete.

In the cloud queuing formulation, the feedback branching is probabilistic in nature. Now given an average file size *Fs* and buffer sizes *Bb*, the probability that the field has been transmitted is now

$$P = \frac{B}{F}$$

Also, the arrival rate at node *WS_2 (A^1)* is the sum of the network arrival rate *(A)* and the rate of the jobs flowing from *In_2* back to *$Ws_{2.}$*

The effect of the framework in fig.1 is that resource management in the context of cloud queuing management will be maximized. With the variance process control and background processes, efficiency in service time, server utilization, queuing response times, queuing workload and contents will be captured for CRCM as explained in fig.2.

2.3 Analysis of Cloud Queuing Model using MATLAB Simevent Tool

MATLAB Simevent tool 2009b was used to develop the cloud queuing model. Considering the cloud based research collaboration model design and fig.1, the following assumptions are made in the model environment as shown in fig.2:

1) The users come from a population that can be considered infinite.
2) Users' arrivals are described by a Poisson distribution with a mean arrival rate of *A* . This means that the time between successive user

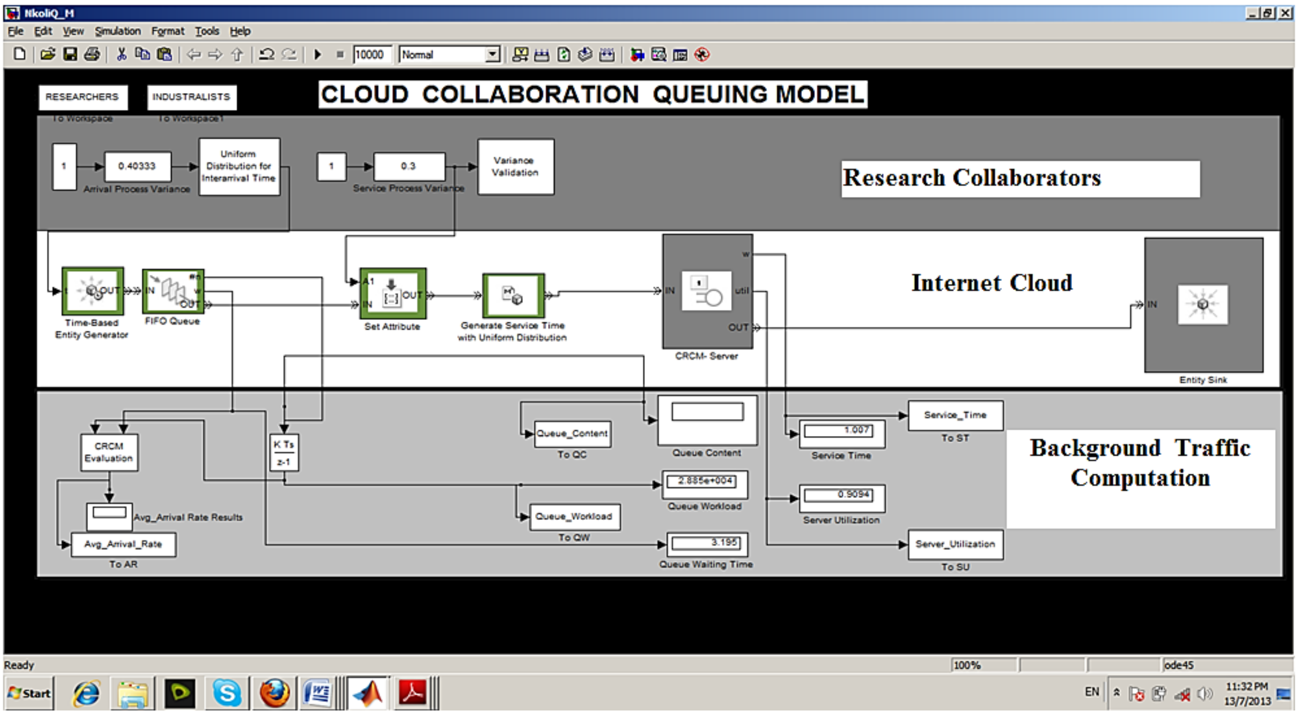

Fig.2 Cloud Collaboration queuing model for CRCM

3) arrivals follows an exponential distribution with an average of $1/A$
4) The users' service rate is described by a Poisson distribution with a mean service rate of $TS\ or\ \mu$(mu). This means that the service time for one customer follows an exponential distribution with an average of $1/\mu$.
5) The waiting line priority rule in the cloud queue CRC used is first-come, first-served.
6) The size of requested files and service times are exponentially distributed.
7) The effects of the HTTP GET requests are negligible since requests are modified for the file processing.
   i. Also, it is assume that the average queue length = (Mean arrival rate)*(Average waiting time in queue)

2.3.1 Description of the Main Blocks of the Cloud Queuing Model

Fig.2 shows the process model of the implemented model showing the Researchers/Industrialist (users) background, the Internet background and the traffic background as well. The model consists of the following main blocks which were configured in it:

- *Arrival Process Variance:* This captures the arrivals, *A* comprising of a low and high limits for *A*. This work used the high margin for A as 0.40333 which is for effective metric studies.
- *Uniform Distribution for Inter-arrival Times*: This output the inter-arrival times according to a uniform distribution with an optimal seed value of 85274. This enables a time-based entity generator for FIFO queue. It creates a signal representing the inter-arrival times for the generated entities. After setting the distribution's variance using the Arrival Process Variance block, the subsystems compute a uniform random variant with the chosen variance and mean.
- *Service Process Variance:* It facilitates the service rate computation by the server comprising of a low and a high limits for TS and was set at 0.3. This was configured to drive the set attribute entity for the incoming FIFO queues *FIFO Queue:* This is based on the blocks available in SimEvent/ Simulink library. The buffer queue is controlled by incoming signals from the uniform distribution for inter arrival times which enables the time based entity generator. This activates the set attribute entity which calls a function; function out_ServiceTime= GenServiceTime which generates uniform service time for the server. The FIFO Queue block is initiated by the arrival of a packet generated by the output of a time based entity generator that synchronizes the arrival of the packet (from users) with the timing signals of the entity generator block. It basically stores entities yet to be served by the server.
- *CRCM Evaluation Block:* Sequential simultaneous arrivals are captured by the CRCM evaluation block which transposes the average arrival results. Here also, the queuing content, workload and waiting time are aggregated.
- *CRCM Server:* These models a server whose service time has a uniform distribution. The operation of this unit is very vital in the model because it processes the service time and the server utilization while outputting the results accordingly in the model. The server block computes the server utilization and average waiting time in the server.
- *Entity Sink:* This accepts or blocks entities and it is found in the SimEvent Library. The block provides a way to terminate an entity path. Selecting Input port

available for entity arrivals, the block always accepts entity arrivals. Neglecting to select input port available for entity arrivals, the block never accepts entity arrivals. The model termination is achieved by this unit.

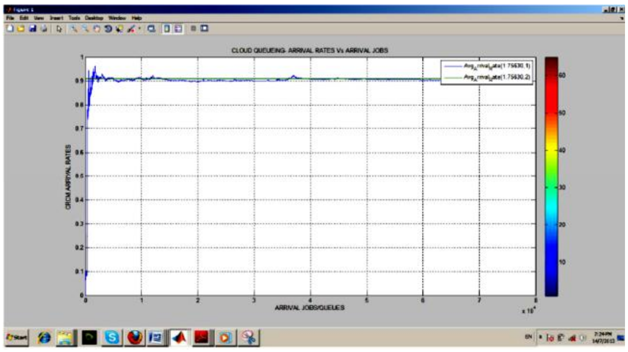

Fig.3 Cloud Queuing Arrival Rate Response

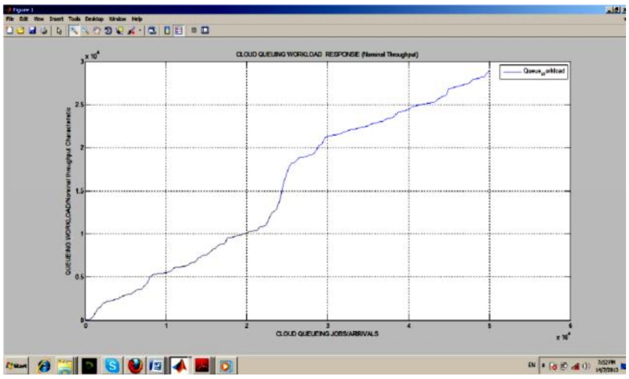

Fig.4 Cloud Queuing Workload/Nominal Throughput Response

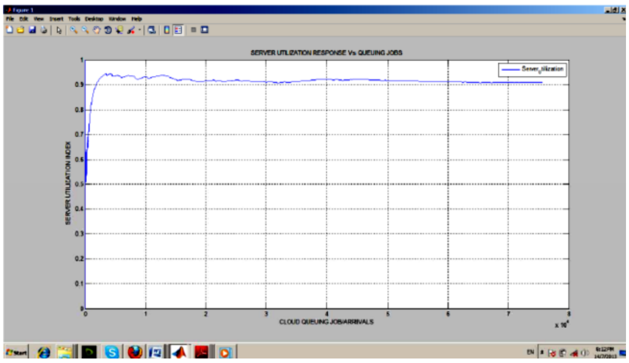

Fig.5 Server Utilization Response

After these configurations, the graphical plots of fig. 2 were obtained via the Simevent command worksheet (see fig. 3) and discussed, showing the effectiveness of fig. 1. From fig. 2, the system response was ascertained by moving the arrival process variance slider or the Service Process Variance slider during the simulation while observing how the queue content changes. When traffic intensity is high, the average waiting time in the queue is approximately linear in the variances of the inter-arrival time and service time as shown in fig. 3. The larger the variances are, the longer an entity has to wait, and the more entities are waiting in the system. The plots of fig. 3 to fig. 7 show the generated results of fig 1.

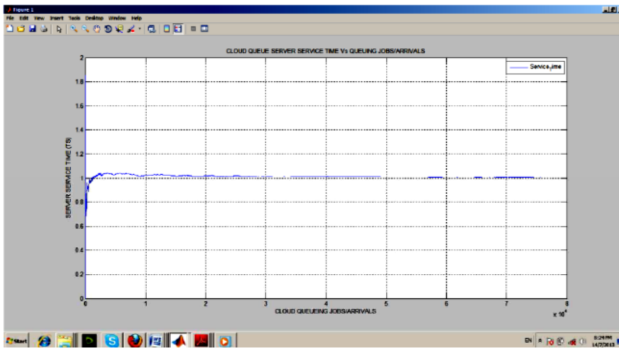

Fig.6 Server Service Time Response

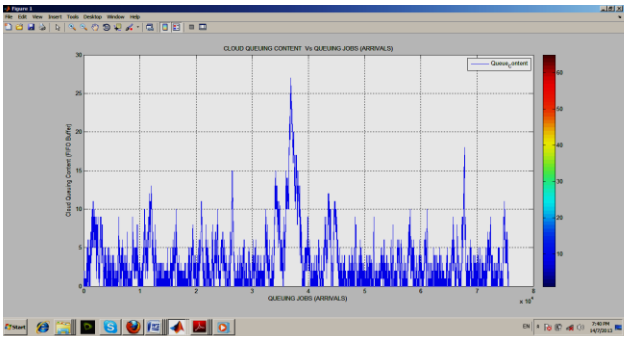

Fig.7 Cloud Queuing Content Response

2.4 Cloud Queuing Model Deductions

Essentially, this work made some interesting deductions from the Cloud collaboration queuing model detailed from fig.3 to fig.6. It was observed from fig.4 that the Queuing workload *QW* which shows a nominal throughput gave 2.88×10^4. Queuing waiting time was observed to be 3.195Sec, Server Utilization was shown in fig.5 to be 0.9094, the service time (TS) was observed to be 1.007sec as shown in fig.6. These were obtained by ensuring that the optimal values of the arrival process variance (0.4033) and service process variance (0.3) were kept constant. This work now proposes these values for the integration phase of the cloud based collaboration platform designed by the authors to facilitate service efficiency and optimal performance.

2.5 CRCM Queuing Content Evaluation

The CRC Http service was configured in the Data centre platform to ascertain its queuing discipline. The notation G/G/1 queue which is usually referred to as generalized single-server queue with First-in-First-out discipline with a general distribution of the sequences of inter-arrival and service times (which are the driving sequences of the system) was observed.

The arrival jobs are modulated by the admission control module (Resource reservation Protocol-RSVP) making for the oscillation in fig. 8. The pattern of this plot is similar to the plot in fig. 7 previously shown. The maximum threshold capacity is about 12.5 which is envisaged to offer efficiency in the collaboration platform.

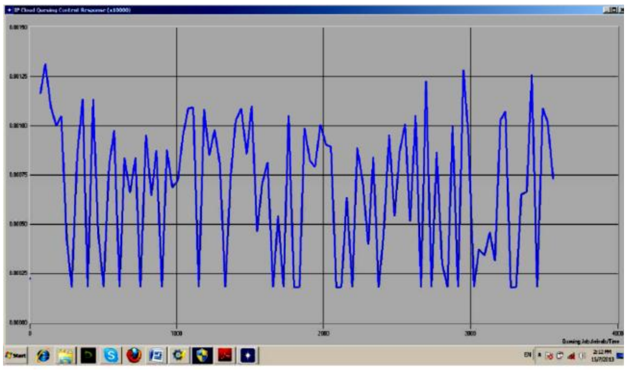

Fig. 8 IP Cloud Queuing Content Response

2.6 CRCM HTTP Queuing Workload

In the CRCM Data centre, Http message queues provide an asynchronous communications protocol, meaning that the sender and receiver of the message do not need to interact with the message queue at the same time. Messages placed onto the queue are stored until the recipient server retrieves them. Message queues have implicit or explicit limits on the size in any workload scenario. It is worthy to note that for the queuing workload, arrivals still occur at rate $A=\lambda$ according to a Poisson process and move the process from state i to $i + 1$.The service times have an exponential distribution as depicted in fig. 9.

Since the buffer is approximated to infinity in the model assumption, there is therefore no limit on the number of jobs it can contain. The plot shown in figure 10 demonstrates that the CRCM will scale efficiently in direct proportion with the incoming job queues or arrivals.

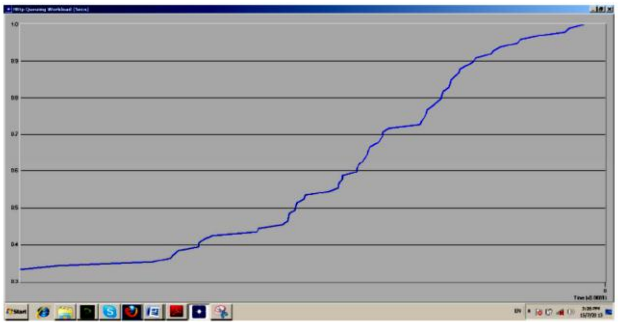

Fig.9 Http Queuing Workload

2.7 Server Utilization Response

Fig. 10 shows the two CRCM Server Utilization Page Response plots for the CRCM shown in the system flow charts. The implication of this figure is that it addresses resource utilization responses elicited from the data centre network, DCN core in the design. Generally, in DCN environment, latency, throughput, server resources utilization, and link bandwidth are vital resources considered in congestion management for collaboration platforms.

With the traffic load sources in the CRCM topology used in this research, an initial gradient was established before resource allocations were fairly distributed.

With a connection request, feasible regions of resource allocation are first established. Resource utilization in the CRCM is very low from the beginning while maintaining a relatively stable state till the end. This implies a fairly uniform resource utilisation by the server. The server utilization is 0.903 which shows a similar trend with fig. 5, hence validating the CRCM design.

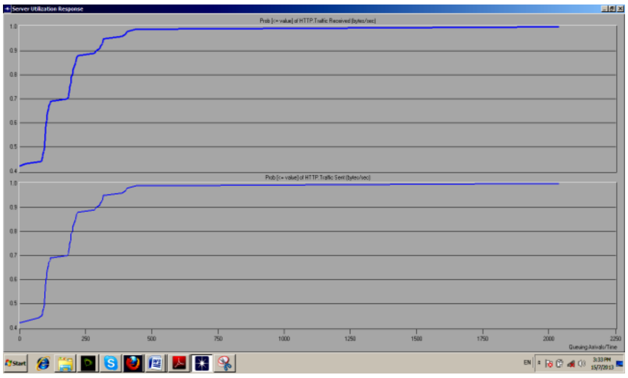

Fig. 10: CRCM Server Utilization Object Response plot

Conclusions

In summary, the deductions from the model for the nominal throughput of the Queuing workload, Queuing waiting time, Server Utilization and service time shows that the model will facilitate service efficiency and optimal performance for the CRCM. The model which was tested and implemented in the actual system shows that queuing theory will optimize network traffic. The model demonstrates that the CRCM will scale efficiently in direct proportion with the incoming job queues or arrivals since the buffer size is approximately infinite.

References

Wikipedia, (Oct. 2013), Queuing theory *http://en.wikipedia.org/wiki/Queueing_theory*

C.K. Leung, (Feb-2000), Queuing Theory and Traffic Engineering, *http://www.ictech.edu.pk/research/MISC/Queuing%20theory%20and%20traffic%20engineering.pdf*

Wang Jian-Ping, Huang Yong, (2008), The monitoring of the network traffic based on queuing theory, *7th International Symposium on Operations Research and Its Applications (ISORA'08),* Lijiang, China, pp 60–65

Nkolika Nwazor, (2014), Modeling a Cloud Computing Based Research Collaboration Platform for Engineering and Technology, Unpublished Ph.D. Dissertation, Computer and Control Systems Engineering, Department of Electronic and Computer Engineering Nnamdi Azikiwe University, Awka.